基于色彩信息的自适应进化点云拼接算法 *

邹　力 [1a, 1b]，葛宝臻 [1a, 1b]，陈　雷 [1a, 1b, 2†]

(1. 天津大学 a. 精密仪器与光电子工程学院; b. 光电信息技术教育部重点实验室，天津 300072; 2. 天津商业大学 信息工程学院，天津 300134)

摘　要：针对现有进化点云拼接算法未使用点云色彩信息的局限性，提出一种基于色彩信息的自适应进化点云拼接算法。使用随机采样与色彩特征点相结合的方式对输入点云进行采样，通过最小化包含色彩约束的点对距离中值建立目标函数，利用自适应进化算法求解两片点云间的最优空间变换，实现点云的有效拼接。对四幅彩色点云进行拼接实验，结果表明，与仅使用空间信息的自适应进化点云拼接算法和其他两种较新的进化拼接算法相比，所提算法在保证同等拼接精度的情况下，能够有效缩短拼接时间。

关键词：彩色点云拼接；自适应进化算法；特征点提取

中图分类号：TP751.1　　doi: 10.3969/j.issn.1001-3695.2017.08.0895

Point cloud registration based on self-adaptive evolutionary optimization and color information

Zou Li[1a, 1b], Ge Baozhen[1a, 1b], Chen Lei[1a, 1b, 2†]

(1. *a. School of Precision Instruments & Opto-Electronics Engineering*, *b. Key Laboratory of Opto-Electronics Information & Technical Science of Ministry of Education*, *Tianjin* 300072, *China*; 2. *School of information Engineering*, *Tianjin University of Commerce*, *Tianjin* 300134, *China*)

Abstract: Traditional evolutionary point cloud registration methods often not using the color information in the models. To overcome the defect, this paper introduced a point cloud registration method based on self-adaptive evolutionary optimization algorithm and color information. It subsampled the input point clouds by extracting the color feature points and randomly chosen points, it utilized the median of all pairs of color constrained points as the object function. At last, it used the self-adaptive evolutionary optimization algorithm to get optimal solution. The registration experiments on four colorized point clouds show that, compared with the evolutionary point cloud registration methods only spatial information use in and two state-of-the-art registration methods, the method significantly shorten the processing time while achieving similar registration precision.

Key Words: color point cloud registration; self-adaptive evolution optimization algorithm; feature points extraction

0　引言

随着三维数字化技术的发展，三维建模已经成为计算机视觉的重要内容，由于三维扫描设备的视野限制以及物体自身复杂的几何形状，实际物体的全部深度信息不可能在一个视点位置获得。为了得到被测物体完整的数据模型，就需要从不同视角对物体进行多次扫描，因此多视角点云的精确拼接是三维建模中的关键一步[1]。

目前应用最广泛的拼接算法是由 Besl 等人[2]提出的迭代最近点（iterative closest point, ICP）算法。ICP 算法通过迭代使得两片点云对应点距离均方误差最小，从而实现点云拼接。ICP 拼接算法速度快，精度较高，但拼接性能易受初始位置、噪声、重合率等因素的影响[2]。近年来，一些学者将进化算法运用于三维点云拼接[3~9]。进化点云拼接算法直接在解空间中寻找最优变换，进化算法的全局搜索性能保证了解不易陷入局部极值，克服了 ICP 算法受初始位置影响的缺点，对噪声较多，重叠率较低的点云也能成功拼接。

然而，已有的进化点云拼接算法主要用于不包含色彩信息的点云拼接。因为传统深度相机和三维扫描仪获得的点云仅仅包含空间位置信息，并不包含表面纹理信息，影响了三维点云的真实感。近年来，一些学者将三维物体彩色信息与深度数据相结合生成三维彩色点云[10~12]。三维彩色点云能提供更好的视觉效果，场景中的物体真实感更强。与此同时，由于点云增加了色彩信息，进化点云拼接算法就有了进一步优化的空间。

基金项目：国家自然科学基金重点项目（61535008）

作者简介：邹力（1993-），男，江苏扬州人，硕士研究生，主要研究方向为进化算法、计算机视觉；葛宝臻（1964-），男，内蒙古卓资人，教授，博士，主要研究方向为光电检测与信息处理、激光粒子测量、激光三维彩色数字化技术；陈雷（1980-），男（通信作者），河北唐山人，博士，副教授，主要研究方向为盲信号处理、智能计算（chenlei@tjcu.edu.cn）.

为此，本文提出一种基于色彩信息的自适应进化点云拼接算法。算法首先通过色彩特征点与随机采样相结合的方式对原始点云进行采样；用包含色彩约束的点对距离中值作为目标函数，该目标函数限制了参与计算的对应点范围，有效降低算法计算量；最后通过自适应进化算法求解两片点云间的最优空间变换。通过对多幅彩色点云的拼接实验表明，与仅使用空间信息的进化点云拼接算法以及两种较新的进化点云拼接算法相比，所提算法在保证收敛精度的前提下，能够在更短的时间内获得更高的拼接成功率。

1　点云拼接背景

点云拼接的目的是要找到一个三维空间中的最佳刚性变换，使得一片点云通过变换后与另一片相邻点云实现对应匹配。一个拼接算法通常包括以下几个部分：

a）输入模型。假设有两片部分重叠的点云，一片为源点云 $P=\{p_i\}$，$i=1,2\cdots N_p$，另一片为目标点云 $Q=\{q_j\}$，$j=1,2\cdots N_q$。p_i 和 q_j 均为三维向量，分别代表空间坐标 X、Y、Z。在彩色点云中，每个点还包括其 RGB 值。通常输入模型的规模很大，直接使用原始点云计算耗时很长。因此需要对源点云进行采样来减少耗时，而目标点云规模保持不变以保证精度[13]。采样点应尽可能均匀的覆盖原始模型，常用的采样方法有随机采样、等距采样和特征点采样等。

b）空间变换 T。它决定了拼接算法的参数空间，主要有刚性变换、仿射变换和非线性变换等。

c）目标函数 $F(T)$。通常点云拼接可以转换为非线性优化问题，目标函数是空间变换 T 的函数，用于量化一个空间变换的优劣程度。

d）优化方法，用于在解空间中求得近似最优解。ICP 是拼接中最常用的优化方法，但是收敛精度受初始位置的影响严重。由于进化算法全局收敛性强、精度高，越来越多学者选择进化算法来优化目标函数。使用进化算法优化目标函数的点云拼接算法称为进化点云拼接算法。

点云拼接的原理如图 1 所示。

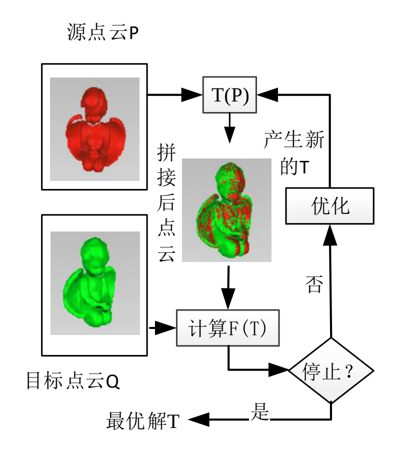


图 1　点云拼接原理

2　自适应进化点云拼接算法

自适应进化点云拼接算法是使用自适应进化算法（self-adaptive evolutionary optimization, SaEvO）[14]作为优化方法的进化点云拼接算法，是目前最优秀的进化点云拼接算法之一。SaEvO 是一种专为点云拼接设计的进化算法，在 2013 年由 Santamaria 等人提出。SaEvO 使用了 meta-optimization 框架，即利用一种进化算法优化另一种优化算法的控制参数。在 SaEvO 中包含两个步骤：a)使用一种基于差分进化[15]和变邻域搜索[16]的进化算法优化目标函数；b)是利用人工免疫算法[17]对步骤 a)中的控制参数进行调节。优化过程中，步骤 a)和 b)交替执行，就可以在最好的控制参数下优化目标函数，从而得到两片点云之间的最佳变换。SaEvO 的算法框架如图 2 所示。

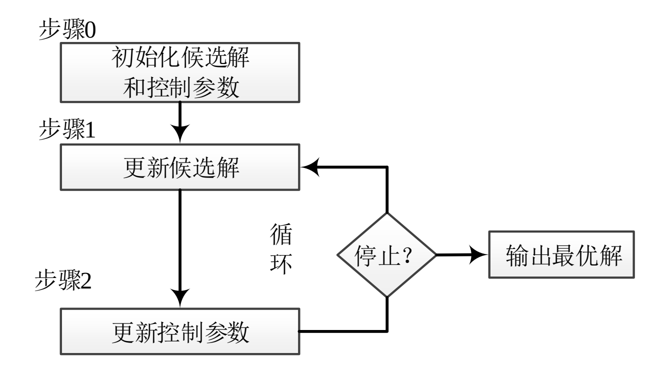


图 2　SaEvO 算法框架

除了优化方法外，自适应进化点云拼接算法的空间变换 T 采用刚性变换，即包括三个绕轴旋转角 (α,β,γ) 和三个平移量 (T_x,T_y,T_z)。通过随机采样方法获得采样点，以点对距离中值建立目标函数 $F(T)$。其计算方法如下：对于源点云 P 中的每一个点 p_i，记 $q_{c(i)}$ 为 p_i 经过 T 变换后在目标点云 Q 中的最近点，通常使用 Kd-tree[18]或者 GCP[19]得到，则对应点距离 d_i 为

$$d_i=\left\|T(p_i)-q_{c(i)}\right\|_2 \tag{1}$$

其中：$T(p_i)$ 表示 p_i 经过 T 变换后的点。则目标函数 $F(T)$ 可以表示为

$$F(T)=MedSE(d_i^2)\quad \forall i=\{1,\ldots,N_p\} \tag{2}$$

其中：$MedSE()$ 表示所有对应点距离平方 d_i^2 的中值。自适应进化点云拼接算法最后通过 SaEvO 优化求解最优空间变换。

在文献[14]中，将 SaEvO 与 Santamaria 提出的分散搜索算法[20]、Silva 提出的改进遗传算法[21]和 Wachowiak 提出的改进粒子群算法[22]等改进进化算法进行了拼接实验对比，结果表明，SaEvO 用于拼接时有优秀的优化求解能力，其自适应的特性使其对不同点云具有很高的鲁棒性，因此也获得了最精确的拼接结果。然而自适应进化点云拼接算法没有使用色彩信息，仅仅针对传统点云的拼接，在彩色点云拼接时还有进一步优化的空间。

3 基于色彩的自适应进化拼接算法

与传统的三维点云不同，三维彩色点云不仅包含了每个点的位置信息，即三维空间坐标 X 、Y 、Z ，还包含了每个点的色彩信息，即 RGB 值。在现有的进化点云拼接算法中没有使用模型的色彩信息，还可以进一步优化。因此本文在点云拼接算法中引入色彩信息，提出一种基于色彩信息的自适应进化点云拼接算法。该算法使用随机采样与色彩特征点结合的采样方法，保证采样点在几何空间和色彩空间均能较均匀的分布；此外，在目标函数中加入匹配点色彩相近的约束，进一步减少了目标函数的算法复杂度；最后通过 SaEvO 求解最优空间变换。

3.1 彩色点云模型及特征点采样

算法中的输入模型为彩色点云模型，每个点不仅包含空间位置信息，还包括 RGB 值。但是不同视角下相同物体，由于受到光照的影响，RGB 值可能相差很大。为了更有效地应用色彩信息，必须消除光照的影响。物体的色调（Hue）受光照的影响较小[23,24]，因此，在本文算法中，用每个点的归一化色调值代表色彩信息，归一化色调值以下简称色调值。RGB 值转换为色调值 H 的公式如下：

$$H=\begin{cases}\dfrac{\theta}{2\pi} & B<=G\\ 1-\dfrac{\theta}{2\pi} & B>G\end{cases} \tag{3}$$

其中：
$$\theta=\arccos\left\{\frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2+(R-B)(G-B)]^{1/2}}\right\} \tag{4}$$

因此，源点云和目标点云可以进一步表示为 $P=\{p_i,h_i^P\}$，$i=1,2\cdots N_p$， $Q=\{q_j,h_j^Q\}$，$j=1,2\cdots N_q$，其中：p_i 和 q_j 均为三维向量，分别代表源点云和目标点云的空间坐标；h_i^P 和 h_j^Q 分别表示 P 中第 i 个点和 Q 中第 j 个点的色调值。

彩色点云拼接中，由于模型色彩分布的不均匀，某种色调比重很大，随机采样可能使大部分采样点均为相同或相近色调，使得算法效率下降。因此需要一种采样方法，使得采样点空间分布均匀，尽可能覆盖整个模型，又色彩分布均匀。为此，提出一种色彩特征点与随机采样相结合的采样方法。首先介绍采样色彩特征点的方法。给定一个参数 n_d ，将色调从 0 到 1 均分为 n_d 份，之后将 P 划分为 n_d 个子点云 P_k ，$k\in 1,2...n_d$ ，每个子点云 P_k 仅包括所有色调 h 满足式（5）的点：

$$\frac{k-1}{n_d}=<h<\frac{k}{n_d} \tag{5}$$

将 Q 分为 n_d 个子点云 Q_w ，$w\in 1,2...n_d$ ，每个子点云 Q_w 仅包括所有色调 h 满足式（6）的点：

$$\frac{w-1.5}{n_d}=<h<\frac{w+0.5}{n_d} \tag{6}$$

$n_d=3$ 时的简单情况下,上述点云划分的示意图如图 3 所示。

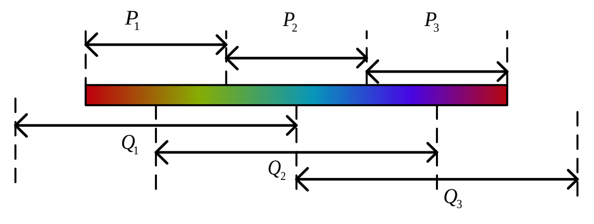


图 3 子点云划分示意图

在 P 的所有子点云中随机选取一个子点云 P_k，若 P_k 不为空且 Q 中相同序号子点云 Q_k 不为空，再在所选子点云 P_k 中随机选取一个点作为色彩特征点。不断重复此过程，即可保证采样点的色调分布基本均匀。设所需色彩特征点数为 N_{sample}，则色彩特征点选取流程如图 4 所示。

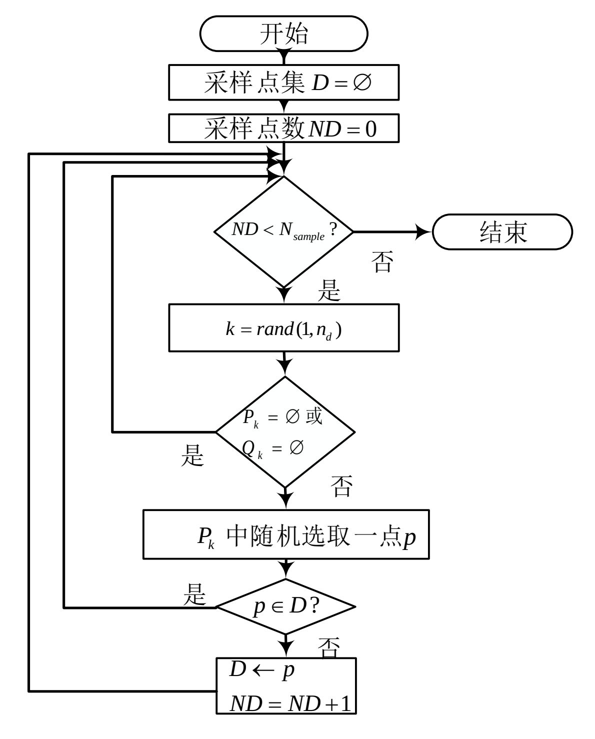


图 4 色彩特征点选取流程

色彩特征点不能保证空间分布均匀，可能造成局部拟合。为了保证采样点空间分布均匀且色彩信息丰富，采用选取色彩特征点和随机采样相结合的方式，首先选出一部分色彩特征点，再随机选取一部分采样点。实际拼接中，取 $n_d=15$ ，色彩特征点与随机采样点的比例为 2:1 为宜。

3.2 带色彩约束的目标函数

本文提出一种带色彩约束的目标函数 $\Psi(T)$。传统距离平方中值目标函数评价一个旋转平移变换时，需要寻找 P 中的每个点 p_i 在 Q 中的最近点 $q_{c(i)}$ 作为匹配点，而寻找匹配点占用了绝大部分运算时间。为了降低查找匹配点的运算时间，提出一种带色彩约束的目标函数，认为匹配点之间的色调相近，所以 p_i 仅仅需要在 Q 中相近色调的点中寻找最近点作为匹配点，不需要搜索所有 Q 中的点。$\Psi(T)$ 增加了匹配点的准确性，色调差异大的点不能成为匹配点，使得函数曲线局部极值点更少，更易于搜索。同时，$\Psi(T)$ 还减少了匹配点的搜索范围，大大缩短了

寻找匹配点的时间。

目标函数 $\Psi(T)$ 中，属于子点云 P_k 的点，仅需要在 Q 中相同序号子点云 Q_k 中查找最近点作为匹配点， Q_k 覆盖的色调范围大于 P_k，这样保证了 P_k 中色调处于边界处的点也能够找到匹配点，不会产生误匹配。色调为 h 的点，所属子点云序号 k 可以用式（7）获得。

$$k=\left\lfloor h/\left(\frac{1}{n_d}\right)\right\rfloor+1 \tag{7}$$

其中：$\lfloor\ \rfloor$ 表示向下取整。可以预先对每一个子点云 Q_k 建立 Kd-tree 数据结构加快最近点搜索。得到匹配点后，用匹配点距离中值作为目标函数值输出。

以下列出了 $\Psi(T)$ 评价空间变换 T 时 $\Psi(T)$ 的操作流程：

for i=1,…, N_p

Step 1) 计算 p_i 经过 T 变换后的点 $T(p_i)$

Step 2) 根据式（7）计算 k

Step 3) 在 Q_k 中寻找最近点 $q_{c(i)}$

Step 4) 计算点对距离 $d_i=\|T(p_i)-q_{c(i)}\|_2$

End

Step 5) 输出结果 $\Psi(T)=MedSE(d_i^2)\quad \forall i=\{1,\cdots,N_p\}$

下面分析目标函数 $\Psi(T)$ 的时间复杂度。假设最近点通过最常用的 Kd-tree 获得，查找最近点的复杂度为 $O(\log N_q)$，则传统中值目标函数的复杂度为 $O(N_p\log N_q)$；假设点云的色调分布均匀，则本文目标函数 Ψ 的复杂度为 $O(N_p\log\frac{N_q}{n_d})$。实际拼接中，点云色调分布不均匀，除了主色调，其他色调比重很低，大部分色彩特征点的对应点很少，因此实际中的算法复杂度将会更低。

3.3 种群编码及优化求解

本文采用 SaEvO 优化求解点云间的最优变换，点云拼接的待求变量为一组空间变换，包括三个旋转角、三个平移向量共六个独立变量。因此采用 $X=[x_1,\ldots,x_6]$ 6 维实值编码，分别代表旋转角（α，β，γ）和平移变量（T_x,T_y,T_z），空间变换矩阵 T 可以表示为

$$T=R_\alpha R_\beta R_\gamma S \tag{8}$$

其中：

$$R_\alpha=\begin{bmatrix}1&0&0&0\\0&\cos\alpha&\sin\alpha&0\\0&-\sin\alpha&\cos\alpha&0\\0&0&0&1\end{bmatrix} \tag{9}$$

$$R_\beta=\begin{bmatrix}\cos\beta&0&-\sin\beta&0\\0&1&0&0\\\sin\beta&0&\cos\beta&0\\0&0&0&1\end{bmatrix} \tag{10}$$

$$R_\gamma=\begin{bmatrix}\cos\gamma&\sin\gamma&0&0\\-\sin\gamma&\cos\gamma&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix} \tag{11}$$

$$S=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\T_x&T_y&T_z&1\end{bmatrix} \tag{12}$$

使用 $\Psi(T)$ 作为目标函数，采样后的源点云与原始的目标点云作为点云的输入，迭代一定时间，即可以获得点云间的近似最优变换。

4 实验结果及分析

为了验证本文算法的有效性，本文共进行了两组拼接实验，实验 1 中将本文算法与仅使用空间信息的进化点云拼接算法（以下简称空间算法）进行了拼接实验对比。两种算法的具体结构如表 1 所示。而在实验 2 中，本文算法将与其他文献中两种较新的进化点云拼接算法进行对比。

表 1 拼接实验一中的对比算法

算法名称	选取采样点	目标函数	优化算法
本文算法	随机采样+色彩特征点	包含色彩约束	SaEvO
空间算法	随机采样	不包含色彩约束	SaEvO

4.1 点云模型

本文共选取了 RGB-D Object Dataset[25]中的四对场景点云作为实验模型，分别为其中的 meeting_small_1_171 和 meeting_small_1_176，简记为 meeting1 和 meeting2；kitchen_small_1_33 和 kitchen_small_1_38，简记为 kitchen1 和 kitchen2；desk_3_27 和 desk_3_32，简记为 desk1 和 desk2；table_1_67 和 table_small_1_72，简记为 table1 和 table2。每片视角的点云包含的点数如表 2 所示。各视角点云如图 5 所示。

表 2 各视角点云规模

模型	点数	模型	点数
meeting1	280915	desk1	275473
meeting2	276579	desk2	255235
kitchen1	279798	table1	288179
kitchen2	270769	table2	286707

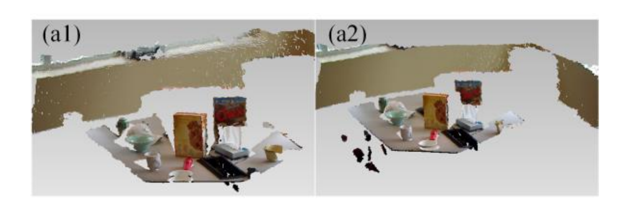
4.2 实验环境

计算机实验平台配置为 Intel Core i7-4790 处理器，主频 3.6 GHz，内存 8 GB，Windows 7 旗舰版 64 位 SP1 操作系统。所有拼接算法均由 C++实现，由 Microsoft Visual Studio2010 编译器编译，最近邻搜索均通过 FLANN 库中的 Kd-tree 实现。

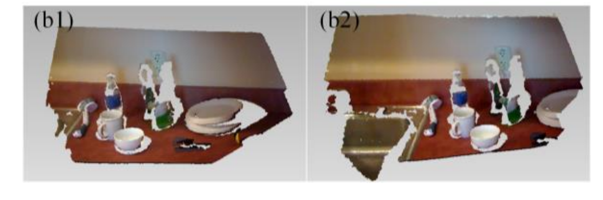
4.3 拼接实验 1 结果分析

实验中，设定了停止条件为运行时间 200 s。由于目标函数不同，不能直接比较收敛结果，因此，每次运行每隔 10 s 记录下当前的最优变换，为了体现公平，最后统一使用传统点对距离中值目标函数评价，为了消除采样点的不同对最终评价的影响，最后统一评价时使用未采样的源点云和目标点云作为输入，并由所得的 20 个目标函数值得到收敛曲线。为了避免随机性，

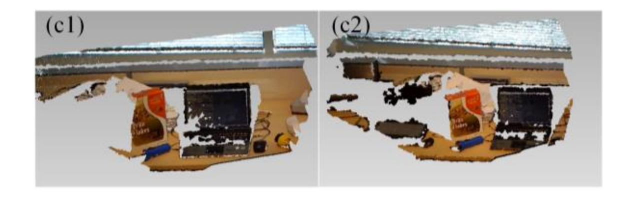
每种算法独立运行 30 次，收敛曲线为 30 次运行的平均曲线。由于优化算法有小概率陷入局部极值，得到的目标函数值远超过正确拼接时的目标函数值，对平均值造成很大的影响，所以本文设定了一个阈值，若目标函数值大于 5e-5，该次运行标记为失败，最终统计 30 次运行中的成功次数，成功时的平均收敛曲线，运行成功时的平均目标函数值和平均迭代次数。



(a1)meeting1　　　　(a2)meeting2



(b1)kitchen1　　　　(b2)kitchen2



(c1)desk1　　　　(c2)desk2



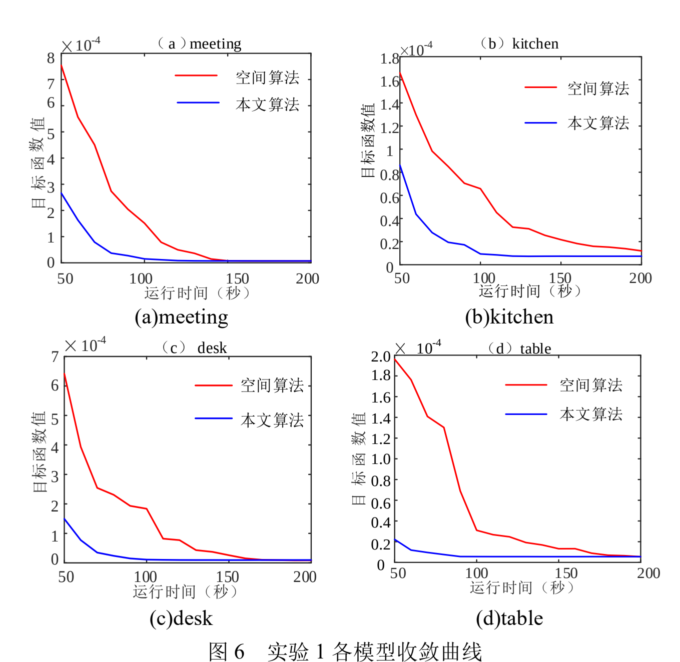
(d1)table1　　　　(d2)table2

图 5　各视角点云

每种算法在源点云中采样 200 个点，采样方法按照表 1 中所示。旋转角度的初始化范围为 $[0,2\pi]$，每维平移向量初始化范围为 $[-1,1]$，自适应进化算法的参数均为种群数量 50，ρ=0.0625，γ=20，β=0.8，ϖ=0.25。

四幅点云的收敛曲线在图 6 中显示。为了避免目标函数值跨度过大，仅绘制出了开始运行 50~200 s 间的收敛曲线。成功运行的平均目标函数值、成功次数及运行平均迭代次数在表 3 中显示。

由图 6 中的收敛曲线可以看出，本文算法的收敛速度明显快于空间算法，一般 100 s 以内即可以达到较好的拼接精度，而空间算法则需要约 180 s 才能够基本收敛。这主要因为本文算法中加入了色彩约束，使得目标函数的复杂度降低。表 3 中的数据能够更好地表现本文算法的优势，本文算法与空间算法的收敛精度大致相同，但是本文算法仅仅在 meeting 模型拼接时有 1 次失败，其他情况均能正确拼接，而空间算法则失败较多，在 desk 模型拼接时就有 5 次失败，总失败次数达到 9 次，陷入局部极值的概率远远高于本文算法，这进一步表现了所提目标函数极值点更少更易于搜索。此外，本文算法 200 s 内迭代次数可以达到 2 500 代左右，而空间算法却只有 1 500 代，仅为本文算法的 60%，可以说明，本文所提目标函数与传统目标函数相比，计算量大大下降。图 7 显示了本文算法拼接成功时的结果。可以看出，两片点云达到了较好的拼接效果。



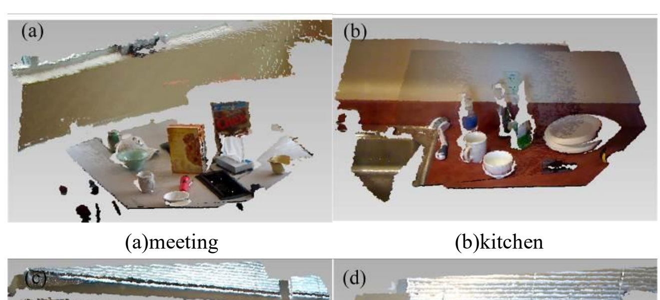
(a)meeting　　　　(b)kitchen

(c)desk　　　　(d)table

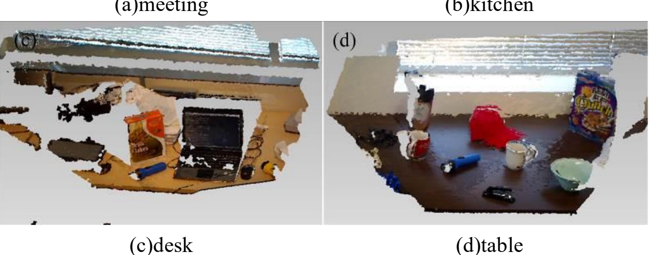
图 6　实验 1 各模型收敛曲线

表 3　实验 1 各模型 30 次运行结果

点云模型	拼接算法	成功次数	平均目标函数值	迭代次数
meeting	本文算法	29	7.24e-6	2552
	空间算法	29	7.09e-6	1488
kitchen	本文算法	30	7.36e-6	2450
	空间算法	29	1.19e-5	1471
desk	本文算法	30	9.90e-6	2567
	空间算法	25	8.72e-6	1512
table	本文算法	30	5.57e-6	2682
	空间算法	28	5.52e-6	1537



(a)meeting　　　　(b)kitchen



(c)desk　　　　(d)table

图 7　点云拼接结果

4.4 拼接实验 2 结果分析

实验 2 中，本文算法将与另外两种进化点云拼接算法进行对比，这两种算法分别为：

a）人工蜂群拼接算法（ABC2016）[8]。2016 年提出，使用 ISS 特征点选取采样点，再使用人工蜂群算法优化求解空间变换。

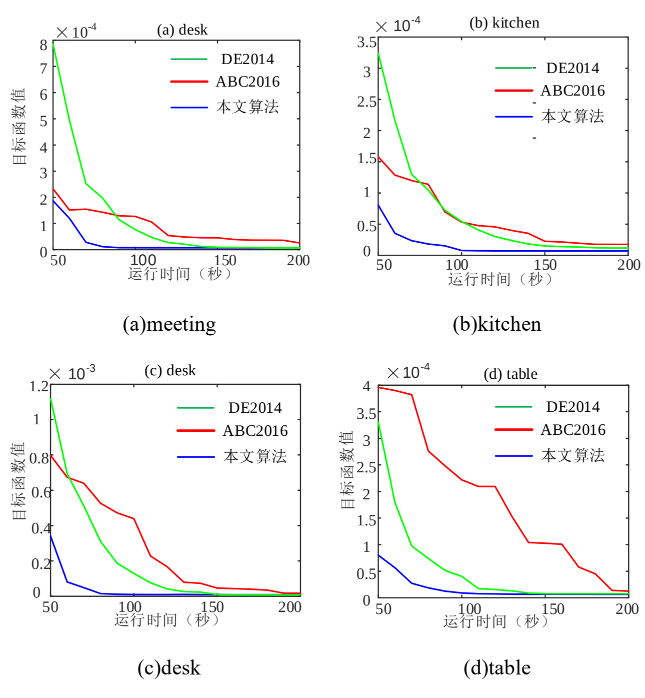
b）侵略差分进化拼接算法（DE2014）[26]。2014 年提出，在差分进化中加入侵略模型。在其原始文献[26]中包括两种策略，本文选择了其中较好的 AIM-dDE-ChAvg 作为对比算法。

实验中，本文算法的控制参数与实验 1 中相同，另外两种算法的控制参数与其原始文献保持一致。各拼接算法的参数如表 4 所示。

表 4 实验 3 算法控制参数

算法名称	控制参数
本文算法	种群数量 50 ρ=0.0625，γ=20，β=0.8，ϖ=0.25
ABC2016	种群数量 20，*limit*=100
DE2014	子种群数目 16，种群规模 10，τ=30

实验 2 中的评价指标、点云模型、实验环境等均与实验一中相同。图 8 显示了实验 2 中各算法的收敛曲线。成功运行的平均目标函数值和成功次数在表 5 中展示。由于每种算法种群数量不同，比较迭代次数意义不大，所以没有记录总迭代次数。



(a)meeting (b)kitchen

(c)desk (d)table

图 8 实验 2 各模型收敛曲线

由图 8 可以发现，本文算法总是有最快的收敛速度，平均比其他算法快 80%到 90%。此外表 5 中本文算法的成功率最高，并且对不同模型的鲁棒性好于其他两种算法。比如 meeting 模型，不管 ABC2016 还是 DE2014 拼接成功率都较低，而本文算法却能够准确拼接。精度方面，本文算法在 kitchen 和 table 模型的拼接中均为最高，其他两种模型本文算法的拼接精度也与最优值相比非常接近。上述实验表明，本文算法相比最新的进化拼接算法收敛速度更快，成功率和精度都很有竞争力。

表 5 实验 2 各模型 30 次运行结果

点云模型	拼接算法	成功次数	平均目标函数值
meeting	本文算法	30	7.49e-6
	ABC2016	10	2.58e-5
	DE2014	20	7.160e-6
kitchen	本文算法	30	7.07e-6
	ABC2016	27	1.76e-5
	DE2014	26	1.17e-5
desk	本文算法	29	9.39e-6
	ABC2016	26	1.72e-5
	DE2014	27	8.93e-6
table	本文算法	30	6.58e-6
	ABC2016	20	1.26e-5
	DE2014	25	7.61e-6

5 结束语

针对彩色点云拼接，提出一种基于色彩信息的自适应进化点云拼接算法。首先采用色彩特征点和随机采样结合的方式对源点云采样，在计算目标函数时，将原始点云划分为多片子点云，将对应点的查找限制在一片子点云中以减少计算量，最后通过自适应进化算法求解最优变换。与仅使用空间信息的自适应进化点云拼接算法相比，本文算法在相同时间内的迭代次数是前者的 1.7 倍，有更快的收敛速度和更高的拼接成功率；与另外两种进化拼接算法相比，本文算法在精度、收敛速度和成功率方面都很有竞争力。证明了本文算法的有效性。

参考文献：

[1] Besl P J, Mckay N D. A method for registration of 3-D shapes [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 1992, 14 (2): 239-256.

[2] Li T, Gao L, Pan Q, et al. Differential evolution algorithm-based range image registration with scaling parameters [C]// Proc of IEEE International Conference on Image Processing. 2016: 4508-4512.

[3] Bermejo E, Cordón O, Damas S, et al. A comparative study on the application of advanced bacterial foraging models to image registration [J]. Information Sciences, 2015, 295: 160-181.

[4] García-Torres J M, Damas S, Cordón O, et al. A case study of innovative population-based algorithms in 3D modeling: artificial bee colony, biogeography-based optimization, harmony search [J]. Expert Systems with Applications, 2014, 41 (4): 1750-1762.

[5] Domeniconi S, Solerio L. Genetic algorithms for voxel-based medical image registration [C]// Proc of IEEE Fourth International Workshop on Computational Intelligence in Medical Imaging. 2013: 22-29.

[6] Falco I D, Cioppa A D, Maisto D, et al. Differential evolution as a viable

tool for satellite image registration [J]. Applied Soft Computing, 2008, 8 (4): 1453-1462.

[7] 葛宝臻，周天宇，陈雷，等．基于改进 ISS 特征点与人工蜂群算法的点云拼接方法 [J]．天津大学学报：自然科学与工程技术版, 2016, 49 (12): 1296-1302.

[8] 高鹏东，彭翔，李阿蒙，等．基于混合遗传算法和点面距离测度的深度像配准 [J]．计算机应用研究, 2007, 24 (12): 354-356, 360.

[9] Khoshelham K, Elberink S O. Accuracy and resolution of kinect depth data for indoor mapping applications [J]. Sensors, 2012, 12 (2): 1437-54.

[10] Nguyen C V, Izadi S, Lovell D. Modeling kinect sensor noise for improved 3D reconstruction and tracking [C]// Proc of the 2nd IEEE International Conference on 3d Imaging, Modeling, Processing, Visualization & Transmission. 2012: 524-530.

[11] 陈晓明，蒋乐天，应忍冬．基于 Kinect 深度信息的实时三维重建和滤波算法研究 [J]．计算机应用研究, 2013, 30 (4): 1216-1218.

[12] Chow K C, Tsui H T, Tong L. Surface registration using a dynamic genetic algorithm [J]. Pattern Recognition, 2004, 37 (1): 105-117.

[13] Santamaría J, Damas S, Cordón O, et al. Self-adaptive evolution toward new parameter free image registration methods [J]. IEEE Trans on Evolutionary Computation, 2013, 17 (4): 545-557.

[14] Storn R, Price K. Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces [J]. Journal of Global Optimization, 1997, 11 (4): 341-359.

[15] Hansen P, Mtadenovic N. Variable neighborhood search [J]. Computers in Operations Research, 1997, 24: 1097-1100.

[16] Dasgupta D, Dasgupta D. Artificial immune systems and their applications [J]. Lecture Notes in Computer Science, 1999, 1 (4): 121-124.

[17] Zhang Z. Iterative point matching for registration of free-form curves and surfaces [J]. International Journal of Computer Vision, 1994, 13 (2): 119-152.

[18] Yamany S M, Ahmed M N, Hemayed E E, et al. Novel surface registration using the grid closest point (GCP) transform [C]// Proc of International Conference on Image Processing. 1998: 809-813.

[19] Santamaría J, Cordón O, Damas S, et al. Performance evaluation of memetic approaches in 3D reconstruction of forensic objects [J]. Soft Computing, 2009, 13 (8): 883-904.

[20] Silva L, Bellon O R, Boyer K L. Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms. [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2005, 27 (5): 762-776.

[21] Wachowiak M P, Smolikova R, Zheng Y, et al. An approach to multimodal biomedical image registration utilizing particle swarm optimization [J]. IEEE Trans on Evolutionary Computation, 2004, 8 (3): 289-301.

[22] Men H, Gebre B, Pochiraju K. Color point cloud registration with 4D ICP algorithm [C]// Proc of IEEE International Conference on Robotics and Automation. 2011: 1511-1516.

[23] Men H, Pochiraju K. Hue-assisted automatic registration of color point clouds [J]. Journal of Computational Design & Engineering, 2014, 1 (4): 223-232.

[24] Lai K, Bo L, Ren X, et al. A large-scale hierarchical multi-view RGB-D object dataset [C]// Proc of IEEE International Conference on Robotics and Automation. 2011: 1817-1824.

[25] Falco I D, Cioppa A D, Maisto D, et al. Using an adaptive invasion-based model for fast range image registration [C]// Proc of GECCO'14 Conference on Genetic and Evolutionary Computation. 2014: 1095-1102.